Author Profiling for Abuse Detection

Pushkar Mishra
Dept. of CS and Technology
University of Cambridge
United Kingdom
pm576@cl.cam.ac.uk

Marco Del Tredici
ILLC
University of Amsterdam
The Netherlands
m.deltredici@uva.nl

Helen Yannakoudakis
Dept. of CS and Technology
The ALTA Institute
University of Cambridge
United Kingdom
hy260@cl.cam.ac.uk

Ekaterina Shutova
ILLC
University of Amsterdam
The Netherlands
e.shutova@uva.nl

Abstract

The rapid growth of social media in recent years has fed into some highly undesirable phenomena such as proliferation of hateful and offensive language on the Internet. Previous research suggests that such abusive content tends to come from users who share a set of common stereotypes and form communities around them. The current state-of-the-art approaches to abuse detection are oblivious to user and community information and rely entirely on textual (i.e., lexical and semantic) cues. In this paper, we propose a novel approach to this problem that incorporates community-based profiling features of Twitter users. Experimenting with a dataset of $16k$ tweets, we show that our methods significantly outperform the current state of the art in abuse detection. Further, we conduct a qualitative analysis of model characteristics. We release our code, pre-trained models and all the resources used in the public domain.

1 Introduction

Abuse, a term used to collectively refer to offensive language, hate speech, sexist remarks, etc., is omnipresent in social media. Users on social media platforms are at risk of being exposed to content that may not only be degrading but also harmful to their mental health in the long term. *Pew Research Center* highlighted the gravity of the situation via a recently released report (Duggan, 2014). As per the report, 40% of adult Internet users have personally experienced harassment online, and 60% have witnessed the use of offensive names and expletives. Expectedly, the majority (66%) of those who have personally faced harassment have had their most recent incident occur on a social networking website or app. While most of these websites and apps provide ways of flagging offensive and hateful content, only 8.8% of the victims have actually considered using such provisions. These statistics suggest that passive or manual techniques for curbing propagation of abusive content (such as flagging) are neither effective nor easily scalable (Pavlopoulos et al., 2017). Consequently, the efforts to automate the detection and moderation of such content have been gaining popularity in natural language processing (NLP) (Waseem and Hovy, 2016; Wulczyn et al., 2017).

Several approaches to abuse detection demonstrate the effectiveness of character-level bag-of-words features in a supervised classification setting (Djuric et al., 2015; Nobata et al., 2016; Davidson et al., 2017). More recent approaches, and currently the best performing ones, utilize recurrent neural networks (RNNs) to transform content into dense low-dimensional semantic representations that are then used for classification (Pavlopoulos et al., 2017; Badjatiya et al., 2017). All of these approaches rely solely on lexical and semantic features of the text they are applied to. Waseem and Hovy (2016) adopted a

Proceedings of the 27th International Conference on Computational Linguistics, pages 1088–1098
Santa Fe, New Mexico, USA, August 20-26, 2018.

more user-centric approach based on the idea that perpetrators of abuse are usually segregated into small demographic groups; they went on to show that gender information of *authors* (i.e., users who have posted content) is a helpful indicator. However, Waseem and Hovy focused only on coarse demographic features of the users, disregarding information about their communication with others. But previous research suggests that users who subscribe to particular stereotypes that promote abuse tend to form communities online. For example, Zook (2012) mapped the locations of racist tweets in response to President Obama's re-election to show that such tweets were not uniformly distributed across the United States but formed clusters instead. In this paper, we present the first approach to abuse detection that leverages author profiling information based on properties of the authors' social network and investigate its effectiveness.

Author profiling has emerged as a powerful tool for NLP applications, leading to substantial performance improvements in several downstream tasks, such as text classification, sentiment analysis and author attribute identification (Hovy, 2015; Eisenstein, 2015; Yang and Eisenstein, 2017). The relevance of information gained from it is best explained by the idea of *homophily*, i.e., the phenomenon that people, both in real life as well as on the Internet, tend to associate more with those who appear similar. Here, similarity can be defined along various axes, e.g., location, age, language, etc. The strength of author profiling lies in that if we have information about members of a community c defined by some similarity criterion, and we know that the person p belongs to c, we can infer information about p. This concept has a straightforward application to our task: knowing that members of a particular community are prone to creating abusive content, and knowing that the author *p* is connected to this community, we can leverage information beyond linguistic cues and more accurately predict the use of abusive/non-abusive language from p. The questions that we seek to address here are: are some authors, and the respective communities that they belong to, more abusive than the others? And can such information be effectively utilized to improve the performance of automated abusive language detection methods?

In this paper, we answer these questions and develop novel methods that take into account community-based profiling features of authors when examining their tweets for abuse. Experimenting with a dataset of $16k$ tweets, we show that the addition of such profiling features to the current state-of-the-art methods for abuse detection significantly enhances their performance. We also release our code (including code that replicates previous work), pre-trained models and the resources we used in the public domain.[1]

2 Related Work

2.1 Abuse detection

Amongst the first ones to apply supervised learning to the task of abuse detection were Yin et al. (2009) who used a linear SVM classifier to identify posts containing harassment based on local (e.g., n-grams), contextual (e.g., similarity of a post to its neighboring posts) and sentiment-based (e.g., presence of expletives) features. Their best results were with all of these features combined.

Djuric et al. (2015) experimented with comments extracted from the Yahoo Finance portal and showed that distributional representations of comments learned using *paragraph2vec* (Le and Mikolov, 2014) outperform simpler bag-of-words (BOW) representations in a supervised classification setting for hate speech detection. Nobata et al. (2016) improved upon the results of Djuric et al. by training their classifier on a combination of features drawn from four different categories: linguistic (e.g., count of insult words), syntactic (e.g., POS tags), distributional semantic (e.g., word and comment embeddings) and BOW-based (word and characters n-grams). They reported that while the best results were obtained with all features combined, character n-grams contributed more to performance than all the other features.

Waseem and Hovy (2016) created and experimented with a dataset of racist, sexist and clean tweets. Utilizing a logistic regression (LR) classifier to distinguish amongst them, they found that character n-grams coupled with gender information of users formed the optimal feature set; on the other hand, geographic and word-length distribution features provided little to no improvement. Working with the same dataset, Badjatiya et al. (2017) improved on their results by training a gradient-boosted decision

[1] https://github.com/pushkarmishra/AuthorProfilingAbuseDetection

tree (GBDT) classifier on averaged word embeddings learnt using a long short-term memory (LSTM) network that they initialized with random embeddings.

Waseem (2016) sampled $7k$ more tweets in the same manner as Waseem and Hovy (2016). They recruited expert and amateur annotators to annotate the tweets as *racism*, *sexism*, *both* or *neither* in order to study the influence of annotator knowledge on the task of hate speech detection. Combining this dataset with that of Waseem and Hovy (2016), Park et al. (2017) explored the merits of a two-step classification process. They first used a LR classifier to separate abusive and non-abusive tweets, followed by another LR classifier to distinguish between racist and sexist ones. They showed that this setup had comparable performance to a one-step classification setup built with convolutional neural networks.

Davidson et al. (2017) created a dataset of about $25k$ tweets wherein each tweet was annotated as being *racist*, *offensive* or *neither of the two*. They tested several multi-class classifiers with the aim of distinguishing clean tweets from racist and offensive tweets while simultaneously being able to separate the racist and offensive ones. Their best model was a LR classifier trained using TF-IDF and POS n-gram features, as well as the count of hash tags and number of words.

Wulczyn et al. (2017) prepared three different datasets of comments collected from the English Wikipedia Talk page; one was annotated for personal attacks, another for toxicity and the third one for aggression. Their best performing model was a multi-layered perceptron (MLP) classifier trained on character n-gram features. Experimenting with the personal attack and toxicity datasets, Pavlopoulos et al. (2017) improved the results of Wulczyn et al. by using a gated recurrent unit (GRU) model to encode the comments into dense low-dimensional representations, followed by a LR layer to classify the comments based on those representations.

2.2 Author profiling

Author profiling has been leveraged in several ways for a variety of purposes in NLP. For instance, many studies have relied on demographic information of the authors. Amongst these are Hovy et al. (2015) and Ebrahimi et al. (2016) who extracted age and gender-related information to achieve superior performance in a text classification task. Pavalanathan and Eisenstein (2015), in their work, further showed the relevance of the same information to automatic text-based geo-location. Researching along the same lines, Johannsen et al. (2015) and Mirkin et al. (2015) utilized demographic factors to improve syntactic parsing and machine translation respectively.

While demographic information has proved to be relevant for a number of tasks, it presents a significant drawback: since this information is not always available for all authors in a social network, it is not particularly reliable. Consequently, of late, a new line of research has focused on creating representations of users in a social network by leveraging the information derived from the connections that they have with other users. In this case, node representations (where nodes represent the authors in the social network) are typically induced using neural architectures. Given the graph representing the social network, such methods create low-dimensional representations for each node, which are optimized to predict the nodes close to it in the network. This approach has the advantage of overcoming the absence of information that the previous approaches face. Among those that implement this idea are Yang et al. (2016), who used representations derived from a social graph to achieve better performance in entity linking tasks, and Chen and Ku (2016), who used them for stance classification.

A considerable amount of literature has also been devoted to sentiment analysis with representations built from demographic factors (Yang and Eisenstein, 2017; Chen et al., 2016). Other tasks that have benefited from social representations are sarcasm detection (Amir et al., 2016) and political opinion prediction (Tălmăcel and Leon, 2017).

3 Dataset

We experiment with the dataset of Waseem and Hovy (2016), containing tweets manually annotated for abuse. The authors retrieved around $136k$ tweets over a period of two months. They bootstrapped their collection process with a search for commonly used slurs and expletives related to religious, sexual, gender and ethnic minorities. From the results, they identified terms and references to entities that

frequently showed up in abusive tweets. Based on this sample, they used a public Twitter API to collect the entire corpus of ca. $136k$ tweets. After having manually annotated a randomly sampled subset of $16,914$ tweets under the categories *racism*, *sexism* or *none* themselves, they asked an expert to review their annotations in order to mitigate against any biases. The inter-annotator agreement was reported at $\kappa = 0.84$, with a further insight that 85% of all the disagreements occurred in the *sexism* class.

The dataset was released as a list of $16,907$ tweet IDs along with their corresponding annotations[2]. Using python's *Tweepy* library, we could only retrieve $16,202$ of the tweets since some of them have now been deleted or their visibility limited. Of the ones retrieved, 1,939 (12%) are labelled as *racism*, 3,148 (19.4%) as *sexism*, and the remaining 11,115 (68.6%) as *none*; this distribution follows the original dataset very closely (11.7%, 20.0%, 68.3%).

We were able to extract community-based information for 1,836 out of the 1,875 unique authors who posted the $16,202$ tweets, covering a cumulative of 16,124 of them; the remaining 39 authors have either deactivated their accounts or are facing suspension. Tweets in the *racism* class are from 5 of the 1,875 authors, while those in the *sexism* class are from 527 of them.

4 Methodology

4.1 Representing authors

In order to leverage community-based information for the authors whose tweets form our dataset, we create an undirected unlabeled community graph wherein nodes are the authors and edges are the connections between them. An edge is instantiated between two authors u and v if u follows v on Twitter or vice versa. There are a total of 1,836 nodes and 7,561 edges. Approximately 400 of the nodes have no edges, indicating *solitary* authors who neither follow any other author nor are followed by any. Other nodes have an average degree[3] of 8, with close to 600 of them having a degree of at least 5. The graph is overall sparse with a density of 0.0075.

From this community graph, we obtain a vector representation, i.e., an embedding that we refer to as *author profile*, for each author using the *node2vec* framework (Grover and Leskovec, 2016). *Node2vec* applies the skip-gram model of Mikolov et al. (2013) to a graph in order to create a representation for each of its nodes based on their positions and their neighbors. Specifically, given a graph with nodes $V = \{v_1, v_2, \ldots, v_n\}$, *node2vec* seeks to maximize the following log probability:

$$\sum_{v \in V} \log Pr\left(N_s(v) \mid v\right)$$

where $N_s(v)$ denotes the *network neighborhood* of node v generated through sampling strategy s.

In doing so, the framework learns low-dimensional embeddings for nodes in the graph. These embeddings can emphasize either their structural role or the local community they are a part of. This depends on the sampling strategies used to generate the neighborhood: if breadth-first sampling (BFS) is adopted, the model focuses on the immediate neighbors of a node; when depth-first sampling (DFS) is used, the model explores farther regions in the network, which results in embeddings that encode more information about the nodes' structural role (e.g., hub in a cluster, or peripheral node). The balance between these two ways of sampling the neighbors is directly controlled by two *node2vec* parameters, namely p and q. The default value for these is 1, which ensures a node representation that gives equal weight to both structural and community-oriented information. In our work, we use the default value for both p and q. Additionally, since *node2vec* does not produce embeddings for *solitary* authors, we map these to a single zero embedding.

Figure 1 shows example snippets from the community graph. Some authors belong to densely-connected communities (left figure), while others are part of more sparse ones (right figure). In either case, *node2vec* generates embeddings that capture the authors' neighborhood.

[2] https://github.com/ZeerakW/hatespeech/blob/master/NAACL_SRW_2016.csv

[3] The degree of a node is equal to the number of its direct connections to other nodes.

(a) Densely-connected authors

(b) Sparsely-connected authors

Figure 1: Snippets from the community graph for our Twitter data.

4.2 Classifying content

We experiment with seven different methods for classifying tweets as one of *racism*, *sexism*, or *none*. We first re-implement three established and currently best-performing abuse detection methods — based on character n-grams and recurrent neural networks — as our baselines. We then test whether incorporating author profiling features improves their performance.

Char n-grams (LR). As our first baseline, we adopt the method used by Waseem and Hovy (2016) wherein they train a logistic regression (LR) classifier on the Twitter dataset using character n-gram counts. We use uni-grams, bi-grams, tri-grams and four-grams, and L_2-normalize their counts. Character n-grams have been shown to be effective for the task of abuse detection (Nobata et al., 2016).

Hidden-state (HS). As our second baseline, we take the "RNN" method of Pavlopoulos et al. (2017) which achieves state-of-the-art results on the Wikipedia datasets released by Wulczyn et al. (2017). The method comprises a 1-layer gated recurrent unit (GRU) that takes a sequence $w_1, \ldots, w_n$ of words represented as d-dimensional embeddings and encodes them into hidden states $h_1, \ldots, h_n$. This is followed by an LR layer that uses the last hidden state h_n to classify the tweet. We make two minor modifications to the authors' original architecture: we deepen the 1-layer GRU to a 2-layer GRU and use softmax instead of sigmoid in the LR layer.[4] Like Pavlopoulos et al., we initialize the word embeddings to GLoVe vectors (Pennington et al., 2014). In all our methods, words not available in the GLoVe set are randomly initialized in the range ± 0.05, indicating the lack of semantic information. By not mapping these words to a single random embedding, we mitigate against the errors that may arise due to their conflation (Madhyastha et al., 2015). A special OOV (out of vocabulary) token is also initialized in the same range. All the embeddings are updated during training, allowing some of the randomly-initialized ones to get task-tuned; the ones that do not get tuned lie closely clustered around the OOV token, to which unseen words in the test set are mapped.

Word-sum (WS). As a third baseline, we adopt the "LSTM+GLoVe+GBDT" method of Badjatiya et al. (2017), which achieves state-of-the-art results on the Twitter dataset we are using. The authors first utilize an LSTM to task-tune GLoVe-initialized word embeddings by propagating the error back from an LR layer. They then train a gradient boosted decision tree (GBDT) classifier to classify texts based on the average of the embeddings of constituent words. We make two minor modifications to this method: we use a 2-layer GRU[5] instead of the LSTM to tune the embeddings, and we train the GBDT classifier on the L_2-normalized sum of the embeddings instead of their average.[6] Although the authors achieved

[4] We also experimented with 1-layer GRU/LSTM and 1/2-layer bi-directional GRUs/LSTMs but performance only worsened or showed no gains; using sigmoid instead of softmax did not have any noteworthy effects on the results either.

[5] We note the deeper 2-layer GRU slightly improves performance.

[6] Although GBDT, as a tree based model, is not affected by the choice of monotonic function, the L_2-normalized sum ensures uniformity of range across the feature set in all our methods.

state-of-the-art results on Twitter by initializing embeddings randomly rather than with GLoVe (which is what we do here), we found the opposite when performing a 10-fold stratified cross-validation (CV). A possible explanation of this lies in the authors' decision to not use stratification, which for such a highly imbalanced dataset can lead to unexpected outcomes (Forman and Scholz, 2010). Furthermore, the authors train their LSTM on the entire dataset (including the test set) without any early stopping criterion, which leads to over-fitting of the randomly-initialized embeddings.

Author profile (AUTH). In order to test whether community-based information of authors is in itself sufficient to correctly classify the content produced by them, we utilize just the author profiles we generated to train a GBDT classifier.

Char n-grams + author profile (LR + AUTH). This method builds upon the LR baseline by appending author profile vectors on to the character n-gram count vectors for training the LR classifier.

Hidden-state + author profile (HS + AUTH) and **Word-sum + author profile (WS + AUTH).** These methods are identical to the *char n-grams + author profile* method except that here we append the author profiling features on to features derived from the *hidden-state* and *word-sum* baselines respectively and feed them to a GBDT classifier.

5 Experiments and Results

5.1 Experimental setup

We normalize the input by lowercasing all words and removing stop words. For the GRU architecture, we use exactly the same hyper-parameters as Pavlopoulos et al. (2017),[7] i.e., 128 hidden units, Glorot initialization, cross-entropy loss, and the Adam optimizer (Kingma and Ba, 2015). Badjatiya et al. (2017) also use the same settings except they have fewer hidden units. In all our models, besides dropout regularization (Srivastava et al., 2014), we hold out a small part of the training set as validation data to prevent over-fitting. We implement the models in *Keras* (Chollet and others, 2015) with *Theano* backend and use 200-dimensional pre-trained GLoVe word embeddings.[8] We employ *Lightgbm* (Ke et al., 2017) as our GDBT classifier and tune its hyper-parameters using 5-fold grid search. For the *node2vec* framework, we use the same parameters as in the original paper (Grover and Leskovec, 2016) except we set the dimensionality of node embeddings to 200 and increase the number of iterations to 25 for better convergence.

5.2 Results

We perform 10-fold stratified cross validation (CV), as suggested by Forman and Scholz (2010), to evaluate all seven methods described in the previous section. Following previous research (Badjatiya et al., 2017; Park and Fung, 2017), we report the average weighted precision, recall, and F_1 scores for all the methods. The average weighted precision is calculated as:

$$\frac{\sum_{i=1}^{10} (w_r \cdot \mathrm{P}_r^i + w_s \cdot \mathrm{P}_s^i + w_n \cdot \mathrm{P}_n^i)}{10}$$

where $\mathrm{P}_r^i, \mathrm{P}_s^i, \mathrm{P}_n^i$ are precision scores on the *racism*, *sexism*, and *none* classes from the i^{th} fold of the CV. The values w_r, w_s, and w_n are the proportions of the *racism*, *sexism*, and *none* classes in the dataset respectively; since we use stratification, these proportions are constant ($w_r = 0.12$, $w_s = 0.19$, $w_n = 0.69$) across all folds. Average weighted recall and F_1 are calculated in the same manner.

The results are presented in Table 1. For all three baseline methods (LR, WS, and HS), the addition of author profiling features significantly improves performance ($p < 0.05$ under 10-fold CV paired t-test). The LR + AUTH method yields the highest performance of $F_1 = 87.57$, exceeding its respective baseline by nearly 4 points. A similar trend can be observed for the other methods as well. These results point to

[7] The authors have not released their models, and we therefore replicate their approach based on the details in their paper.

[8] http://nlp.stanford.edu/data/glove.twitter.27B.zip

the importance of community-based information and author profiling in abuse detection and demonstrate that our approach can further improve the performance of existing state-of-the-art methods.

	Method	P	R	F_1
Baselines	LR	84.07	84.31	83.81
	HS	83.50	83.71	83.54
	WS	82.86	83.10	82.37
Our methods	AUTH	72.13	76.05	71.26
	LR + AUTH	**87.57**	**87.66**	**87.57**
	HS + AUTH	87.29	87.32	87.29
	WS + AUTH	87.11	87.20	87.08

Table 1: Average weighted precision, recall and F_1 scores of the different methods on the Twitter datasest. All improvements are significant ($p < 0.05$) under 10-fold CV paired t-test.

Method	P	R	F_1
LR	**77.29**	67.92	72.28
HS	74.15	72.46	73.24
WS	76.43	67.77	71.78
AUTH	43.33	0.31	0.61
LR + AUTH	76.10	**74.16**	**75.09**
HS + AUTH	74.42	73.54	73.91
WS + AUTH	75.12	72.46	73.72

(a) *Racism* class

Method	P	R	F_1
LR	82.66	63.98	72.09
HS	76.04	68.84	72.24
WS	81.75	57.37	67.38
AUTH	66.85	75.44	70.88
LR + AUTH	86.22	79.07	82.47
HS + AUTH	84.15	**81.32**	**82.75**
WS + AUTH	**86.37**	77.92	81.91

(b) *Sexism* class

Table 2: Performance of the methods on the *racism* and *sexism* classes separately. All improvements are significant ($p < 0.05$) under 10-fold CV paired t-test.

In Table 2, we further compare the performance of the different methods on the *racism* and *sexism* classes individually. As in the previous experiments, the scores are averaged over 10 folds of CV. Of particular interest are the scores for the *sexism* class where the F_1 increases by over 10 points upon the addition of author profiling features. Upon analysis, we find that such a substantial increase in performance stems from the fact that many of the 527 unique authors of the sexist tweets are closely connected in the community graph. This allows for their penchant for sexism to be expressed in their respective author profiles.

The author profiling features on their own (AUTH) achieve impressive results overall and in particular on the *sexism* class, where their performance is typical of a community-based generalization, i.e., low precision but high recall. For the *racism* class on the other hand, the performance of AUTH on its own is quite poor. This contrast can be explained by the fact that tweets in the *racism* class come from only 5 unique authors who: (i) are isolated in the community graph, or (ii) have also authored several tweets in the *sexism* class, or (iii) are densely connected to authors from the *sexism* and *none* classes which possibly camouflages their racist nature.

We believe that the gains in performance will be more pronounced as the underlying community graph grows since there will be less solitary authors and more edges worth harnessing information from.[9] Even when the data is skewed and there is an imbalance of abusive vs. non-abusive authors, we do expect our approach to still be able to identify clusters of authors with similar views.

6 Analysis and discussion

We conduct a qualitative analysis of system errors and the cases where author profiling leads to the correct classification of previously misclassified examples. Table 3 shows examples of abusive tweets from the dataset that are misclassified by the LR method, but are correctly classified upon the addition of author profiling features, i.e., by the LR + AUTH method. It is worth noting that some of the wins scored by the latter are on tweets that are part of a larger abusive discourse or contain links to abusive

[9]Regarding the scalability of our approach, we quote the authors of *node2vec*: "The major phases of *node2vec* are trivially parallelizable, and it can scale to large networks with millions of nodes in a few hours".

content while not explicitly having textual cues that are indicative of abuse per se. The addition of author profiling features may then be viewed as a proxy for wider discourse information, thus allowing us to correctly resolve the cases where lexical and semantic features alone are insufficient.[10]

Tweet	Predicted label	
	LR	LR + AUTH
@Mich_McConnell Just "her body" right?	*none*	*sexism*
@Starius: #GamerGate https://t.co/xuFwsIgxFK WE WIN! ahahahaha	*none*	*sexism*
#Islam dominates our crime, prison & welfare system & national security. Why are we still importing it? @PeterDutton_MP #amagenda #auspol	*none*	*racism*
@Wateronatrain: @MT8_9 You might like this #patriarchy http://t.co/c9m2pFmFJ3	*none*	*sexism*
It seems that Allah sits around all day obsessing about women's hands and faces showing. I guess idiots need a god on their level. #Islam	*none*	*racism*
@SalemP08: @MT8_9 @LiljaOB @midnitebacon @Superjutah @Transic_nyc her response is pretty terrifying.	*none*	*sexism*
@JosephIsVegan @SumbelinaZ @IronmanL1 @Hatewatch Why would you profile white people. Blacks murder at 6 times the rate as whites.	*none*	*racism*

Table 3: Examples of improved classification upon the addition of author profiling features (AUTH).

However, a number of abusive tweets still remain misclassified despite the addition of author profiling features. According to our analysis, many of these tend to contain URLs to abusive content, e.g., "*@salmonfarmer1: Logic in the world of Islam http://t.co/6nALv2HPc3*" and "*@juliarforster Yes. http://t.co/ixbt0uc7HN*". Since Twitter shortens all URLs into a standard format, there is no indication of what they refer to. One way to deal with this limitation could be to additionally maintain a blacklist of links. Another source of system errors is the deliberate obfuscation of words by authors in order to evade detection, e.g., "*Kat, a massive c*nt. The biggest ever on #mkr #cuntandandre*". Current abuse detection methods, including ours, do not directly attempt to address this issue. While this is a challenge for bag-of-word based methods such as LR, we hypothesize that neural networks operating at the character level may be helpful in recognizing obfuscated words.

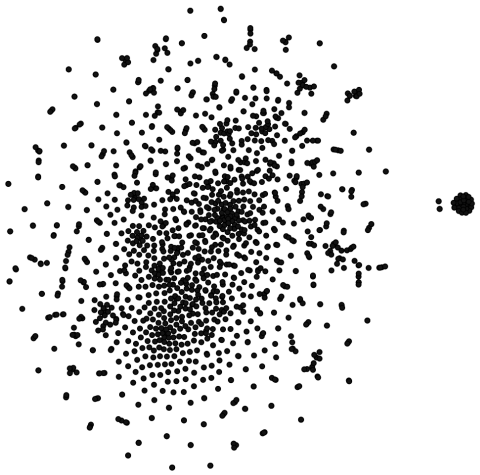

Figure 2: Visualization of author embeddings in 2-dimensional space.

We further conducted an analysis of the author embeddings generated by *node2vec*, in order to validate that they capture the relevant aspects of the community graph. We visualized the author embeddings in

[10]We note that the annotators of the dataset took discourse into account when annotating the tweets. However, the dataset was released as a list of tweet ID and corresponding annotation (racism/sexism/none) pairs; there is no annotation available regarding which tweets are related to which other ones.

(a) *None* class (b) *Sexism* class

Figure 3: Visualization of authors from different classes.

2-dimensional space using *t*-SNE (van der Maaten and Hinton, 2008), as shown in Figure 2. We observe that, as in the community graph, there are a few densely populated regions in the visualization that represent authors in closely knit groups who exhibit similar characteristics. The other regions are largely sparse with smaller clusters. Note that we exclude *solitary* users from this visualization since we have to use a single zero embedding to represent them.

Figure 3 further provides visualizations for authors from the *sexism* and *none* classes separately. While the authors from the *none* class are spread out in the embedding space, the ones from the *sexism* class are more tightly clustered. Note that we do not visualize the 5 authors from the *racism* class since 4 of them are already covered in the *sexism* class.

7 Conclusions

In this paper, we explored the effectiveness of community-based information about authors for the purpose of identifying abuse. Working with a dataset of $16k$ tweets annotated for *racism* and *sexism*, we first comprehensively replicated three established and currently best-performing abuse detection methods based on character n-grams and recurrent neural networks as our baselines. We then constructed a graph of all the authors of tweets in our dataset and extracted community-based information in the form of dense low-dimensional embeddings for each of them using *node2vec*. We showed that the inclusion of author embeddings significantly improves system performance over the baselines and advances the state of the art in this task. Users prone to abuse do tend to form social groups online, and this stresses the importance of utilizing community-based information for automatic abusive language detection.

In the future, we wish to explore the effectiveness of community-based author profiling in other tasks such as stereotype identification and metaphor detection.

References

Silvio Amir, Byron C Wallace, Hao Lyu, and Paula Carvalho Mário J Silva. 2016. Modelling context with user embeddings for sarcasm detection in social media. *arXiv preprint arXiv:1607.00976*.

Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. 2017. Deep learning for hate speech detection in tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion*, WWW '17 Companion, pages 759–760, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

Wei-Fan Chen and Lun-Wei Ku. 2016. Utcnn: a deep learning model of stance classification on social media text. *arXiv preprint arXiv:1611.03599*.

Huimin Chen, Maosong Sun, Cunchao Tu, Yankai Lin, and Zhiyuan Liu. 2016. Neural sentiment classification with user and product attention. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1650–1659.

François Chollet et al. 2015. Keras.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the 11th International AAAI Conference on Web and Social Media*, ICWSM '17.

Nemanja Djuric, Jing Zhou, Robin Morris, Mihajlo Grbovic, Vladan Radosavljevic, and Narayan Bhamidipati. 2015. Hate speech detection with comment embeddings. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15 Companion, pages 29–30, New York, NY, USA. ACM.

Maeve Duggan. 2014. Online harassment.

Javid Ebrahimi and Dejing Dou. 2016. Personalized semantic word vectors. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 1925–1928. ACM.

Jacob Eisenstein. 2015. Written dialect variation in online social media. *Charles Boberg, John Nerbonne, and Dom Watt, editors, Handbook of Dialectology. Wiley*.

George Forman and Martin Scholz. 2010. Apples-to-apples in cross-validation studies: Pitfalls in classifier performance measurement. *SIGKDD Explor. Newsl.*, 12(1):49–57, November.

Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Dirk Hovy. 2015. Demographic factors improve classification performance. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 752–762.

Anders Johannsen, Dirk Hovy, and Anders Søgaard. 2015. Cross-lingual syntactic variation over age and gender. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 103–112.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 3149–3157. Curran Associates, Inc.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, ICLR '15.

Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. *CoRR*, abs/1405.4053.

Pranava Swaroop Madhyastha, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2015. Mapping unseen words to task-trained embedding spaces. *CoRR*, abs/1510.02387.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Shachar Mirkin, Scott Nowson, Caroline Brun, and Julien Perez. 2015. Motivating personality-aware machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1102–1108.

Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. 2016. Abusive language detection in online user content. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, pages 145–153, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

Ji Ho Park and Pascale Fung. 2017. One-step and two-step classification for abusive language detection on twitter. In *Proceedings of the First Workshop on Abusive Language Online*, pages 41–45. Association for Computational Linguistics.

Umashanthi Pavalanathan and Jacob Eisenstein. 2015. Confounds and consequences in geotagged twitter data. *arXiv preprint arXiv:1506.02275*.

John Pavlopoulos, Prodromos Malakasiotis, and Ion Androutsopoulos. 2017. Deep learning for user comment moderation. In *Proceedings of the First Workshop on Abusive Language Online*, pages 25–35. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Ciprian Tălmăcel and Florin Leon. 2017. Predicting political opinions in social networks with user embeddings. In *Proceedings of the 2017 13th IEEE International Conference on Intelligent Computer Communication and Processing*.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605.

Zeerak Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL Student Research Workshop*, pages 88–93, San Diego, California, June. Association for Computational Linguistics.

Zeerak Waseem. 2016. Are you a racist or am i seeing things? annotator influence on hate speech detection on twitter. In *Proceedings of the First Workshop on NLP and Computational Social Science*, pages 138–142. Association for Computational Linguistics.

Ellery Wulczyn, Nithum Thain, and Lucas Dixon. 2017. Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, pages 1391–1399, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

Yi Yang and Jacob Eisenstein. 2017. Overcoming language variation in sentiment analysis with social attention. *Transactions of the Association for Computational Linguistics*.

Yi Yang, Ming-Wei Chang, and Jacob Eisenstein. 2016. Toward socially-infused information extraction: Embedding authors, mentions, and entities. *arXiv preprint arXiv:1609.08084*.

Dawei Yin, Brian D. Davison, Zhenzhen Xue, Liangjie Hong, April Kontostathis, and Lynne Edwards. 2009. Detection of harassment on web 2.0. In *Processings of the Content Analysis in the WEB 2.0*, 2:1-7.

Matthew Zook. 2012. Mapping racist tweets in response to president obama's re-election. [Online; accessed 15 March 2018].